UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/781,525	02/08/2001	Shahrokh Sadjadi	50325-0510	2683

29989 7590 02/26/2003

HICKMAN PALERMO TRUONG & BECKER, LLP
1600 WILLOW STREET
SAN JOSE, CA 95125

EXAMINER
PHAM, HUNG Q

ART UNIT	PAPER NUMBER
2172	

DATE MAILED: 02/26/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/781,525	SADJADI, SHAHROKH	
	Examin r	Art Unit	
	HUNG Q PHAM	2172	

-- The MAILING DATE of this communicati n appears on the cover sheet with the corresp ndence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____ .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-17* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-7 and 13-17* is/are rejected.

7)☒ Claim(s) *8-12* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☒ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other:

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-7, 10, and 13-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kavanagh et al. [USP 5,742,813] in view of Vahalia et al. [USP 6,389,420 B1].

Regarding to claims 1, 13, 15, and 17, Kavanagh teaches a method, a computer program, and an apparatus for concurrency controlling a plurality of users in an object oriented database management system, and allowing editing of the database while other users are concurrently searching the database by using a client/server architecture comprising a knowledge base client and a knowledge base server (Kavanagh, Abstract). The object oriented database management system is a client/server database system with a dedicated server manages the entire database, and all other nodes are sometimes referred to as clients. The clients communicate with the server to access the database on behalf of the applications that run on them (Kavanagh, Col. 1, lines 15-29). The Kavanagh method provides optimal availability by

allowing users to query and view class objects without disruption of their view while modifications such as additions, deletions, and edits of classes, attributes, instances, and parameters are being made by other users (Kavanagh, Col. 3, lines 49-58), and implements concurrency control in an object-oriented database using three types of lock modes: class share lock, tree update lock, tree exclusive lock and update lock, which is used for certain actions including modifying parameter values, adding, and moving instances (Kavanagh, Col. 8, line 65-Col. 9, line 45). As shown in FIGS. 10-12 are the flow diagrams representing the steps of a process that occurs when a user selects the "find class" activity. FIG. 14 is a diagram of a schema 248 corresponding to the display of FIG. 13, and it illustrates corresponding internal lock states of the classes 245, 240, 241, 246, 243, and 247 in the schema 248. FIG. 15 illustrates a lock table 250 as ***a lock data structure*** maintained by the lock manager 125 and corresponds to the schema 248 depicted in FIG. 14 and displayed in FIG. 13. The rows identified by reference numerals 251, 252, 253, 254, and 255 of the lock table 250 each corresponds to a class 245, 240, 241, 246, 243, and 247, respectively, in the schema 248 as ***data indicative of values for a resource object identification***. Each lock holder has a corresponding column as lock objects 256, 257, 258, and 259. Class handle 251 in the lock table 250 has a CSL lock object 261 as ***lock type*** associated with lock holder 257 because the class 245 in the schema 248 is open on the display 116 of the user who is lock holder 257. The class 241 in the schema 248 has a CSL 262 because the user who is lock holder 257 also has it open. Class 243 in the schema 248 has a CSL lock object 260 because it is the selected class (Kavanagh, Col. 18, line 4-Col. 19, line 26). The Kavanagh technique as

discussed indicates the step of ***creating and storing a lock data structure for a particular resource object, the lock data structure comprising data indicative of values for a resource object identification, and a lock type***. As shown in FIG. 38, the steps that are involved in concurrency control when using the schema editor to change the structure of the schema is described. In step 340 when the user selects the schema developer or schema editor 144 for obtaining a TXL lock on the sub-tree that the user wishes to modify at step 341, where a tree exclusive lock is requested for the active class 243. If the TXL cannot be obtained, then the process branches to step 342 and the schema developer 144 cannot be started. When the TXL lock is granted, the method proceeds to step 343 and the schema developer screen 350 is displayed. After obtaining a CSL lock by the schema developer 144 for the parent class 241 of the class 243, the schema could be edited in step 345 (Kavanagh, Col. 23, lines 17-35). The technique as disclosed in FIG. 38 indicates the steps of ***receiving a request from a requesting process for a requested lock type for access to the particular resource object; and determining whether to grant the request based on the requested lock type and the lock type in the lock data structure.*** Kavanagh does not disclose the lock data structure has ***a version number related to a number of changes to the resource object since the lock data structure was generated***. Vahalia teaches a method for distributing file locks and file metadata from a file manager to clients in a data network to permit the clients to share access to file data in data storage (Vahalia, Abstract). Vahalia further discloses a version number associated with the metadata of each file is used to guarantee that every client or file manager accessing a file always uses the most up-to-date version of the metadata. Every time the metadata

is changed on a client or file manager, the version number associated with that metadata on that client or file manager is increased by one. To avoid a data security problem, the metadata in the file system is always written back to data storage after the corresponding data has been updated (Vahalia, Col. 17, lines 24-45). The Vahalia technique as discussed indicates ***a version number related to a number of changes to the resource object*** since the metadata was generated. Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the Kavanagh method by including a version number in the lock table, and by doing this, a user will access a most up-to-date version of the class and class object when a specified lock is granted.

Regarding to claim 2, Kavanagh and Vahalia teaches all the claimed subject matters as discussed in claim 1, Kavanagh further discloses the steps of ***bringing the value of the lock type in the data structure into agreement with the lock type in the request; generating a lock object having data indicative of the values for the resource object identification, the lock type and the version number from the lock data structure, and returning the lock to the requesting process*** (Kavanagh, FIG. 38-39, Col. 23).

Regarding to claim 3, Kavanagh and Vahalia teaches all the claimed subject matters as discussed in claim 1, Kavanagh further discloses the steps of ***receiving a lock to be released having data indicative of values for the resource object identification and the lock type and the version number; determining whether the data indicative of the value for the***

lock type in the lock to be released indicates an exclusive lock, and if it is determined the data indicates the exclusive lock is to be released, then changing the value for the version number in the lock data structure based on the value of the version number in the lock to be released (Kavanagh, FIG. 38-39, Col. 23; and Vahalia, Col. 17, lines 24-45).

Regarding to claim 4, Kavanagh and Vahalia teaches all the claimed subject matters as discussed in claim 2, Kavanagh further discloses the step of ***the lock data structure further comprises a reference number; said step of generating a lock data structure further comprises setting the reference number to a predetermined initial value; and said method further comprises, if it is determined to grant the request, then replacing the value of the reference number in the lock data structure with a sum of the value of the reference number in the lock data structure and a predetermined reference change value*** (FIG. 15).

Regarding to claim 5, Kavanagh and Vahalia teaches all the claimed subject matters as discussed in claim 4, Kavanagh further discloses the step of ***receiving a lock to be released having data indicating the particular resource object; determining whether the reference number substantially equals the predetermined initial value of the reference number; and if it is determined the reference number does not substantially equal the predetermined initial value, then replacing the value of the reference number in the lock data structure with a difference substantially equal to the value of the reference number in the lock data structure minus the predetermined reference change*** (FIG. 15).

Regarding to claim 6, Kavanagh and Vahalia teaches all the claimed subject matters as discussed in claim 5, Kavanagh further discloses the step of ***deleting the lock data structure for the particular resource object if it is determined the reference substantially equals the predetermined initial value*** (Kavanagh, Col. 10, lines 20-28).

Regarding to claims 7, 14, and 16, Kavanagh teaches a method, a computer program, and an apparatus for concurrency controlling a plurality of users in an object oriented database management system, and allowing editing of the database while other users are concurrently searching the database by using a client/server architecture comprising a knowledge base client and a knowledge base server (Kavanagh, Abstract). The object oriented database management system is a client/server database system with a dedicated server manages the entire database, and all other nodes are sometimes referred to as clients. The clients communicate with the server to access the database on behalf of the applications that run on them (Kavanagh, Col. 1, lines 15-29). The Kavanagh method provides optimal availability by allowing users to query and view class objects without disruption of their view while modifications such as additions, deletions, and edits of classes, attributes, instances, and parameters are being made by other users (Kavanagh, Col. 3, lines 49-58), and implements concurrency control in an object-oriented database using three types of lock modes: class share lock, tree update lock, tree exclusive lock and update lock, which is used for certain actions including modifying parameter values, adding, and moving instances (Kavanagh, Col. 8, line 65-Col. 9, line 45). As shown in FIG. 38, the steps that

are involved in concurrency control when using the schema editor to change the structure of the schema is described. In step 340 when the user selects the schema developer or schema editor 144 for obtaining a TXL lock on the sub-tree that the user wishes to modify at step 341, where a tree exclusive lock is requested for the active class 243. If the TXL cannot be obtained, then the process branches to step 342 and the schema developer 144 cannot be started. When the TXL lock is granted, the method proceeds to step 343 and the schema developer screen 350 is displayed. After obtaining a CSL lock by the schema developer 144 for the parent class 241 of the class 243, the schema could be edited in step 345 (Kavanagh, Col. 23, lines 17-35). The technique as disclosed in FIG. 38 indicates the steps of ***receiving from a client process a request to update a particular resource object; sending to a lock manager process a request for a first lock for access to the particular resource object, the request including data indicating an optimistic lock type; receiving the first lock for access to the particular resource object; using the optimistic lock to update the resource object***. As shown in FIG. 39 is a lock table 250 that indicates the locks that are held during the operations as in FIG. 38. The rows identified by reference numerals 251, 252, 253, 254, and 255 of the lock table 250 each corresponds to a class 245, 240, 241, 246, 243, and 247, respectively, in the schema 248 as ***data indicative the resource object***. Each lock holder has a corresponding column as lock objects 256, 257, 258, and 259. Class handle 255 in the lock table 250 has a TXL lock object 260 as ***the optimistic lock type*** (Kavanagh, Col. 18, line 57-Col. 19, line 26). The technique as disclosed in FIG. 39 indicates ***the first lock including data indicating the resource object,*** and ***the optimistic lock type***. Kavanagh does not disclose the

lock data structure has ***a first value for a version number related to a number of changes to the resource object since the lock manager generated a lock data structure corresponding to the resource object***. Vahalia teaches a method for distributing file locks and file metadata from a file manager to clients in a data network to permit the clients to share access to file data in data storage (Vahalia, Abstract). Vahalia further discloses a version number associated with the metadata of each file is used to guarantee that every client or file manager accessing a file always uses the most up-to-date version of the metadata. Every time the metadata is changed on a client or file manager, the version number associated with that metadata on that client or file manager is increased by one. To avoid a data security problem, the metadata in the file system is always written back to data storage after the corresponding data has been updated (Vahalia, Col. 17, lines 24-45). The Vahalia technique as discussed indicates ***a first value for a version number related to a number of changes to the resource object*** since the metadata was generated. Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the Kavanagh method by including a version number in the lock table, and by doing this, a user will access a most up-to-date version of the class and class object when a specified lock is granted.

Regarding to claim 10, Kavanagh and Vahalia teaches all the claimed subject matters as discussed in claim 7, Kavanagh further discloses the step of ***sending to the lock manager process a first release message to release the first lock*** (Kavanagh, FIG. 38).

Allowable Subject Matter

3. Claims 8, 9, 11, and 12 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Regarding to claim 8, 9, and 12, Kavanagh and Vahalia teaches all the claimed subject matters as discussed in claim 7, but the Kavanagh and Vahalia prior art does not update a resource object by ***sending to a lock manager process a request for a second lock for access to the particular resource object, the request including data indicating the resource object identification and an exclusive lock type; receiving the second lock for access to the particular resource object, the second lock including data indicating the resource object identification, the exclusive lock type and a second value for the version number; determining whether the second value for the version number substantially equals the first value for the version number; and if the second value substantially equals the first value, then committing an updated resource object to the resource, and replacing the second value in the reference number in the second lock with a third value of the version number, the third value computed by adding the second value and a predetermined version change value; and if the second value does not substantially equal the first value, then sending a message to the client process, the message indicating that the resource object was not updated; sending to the lock manager process a second release message to release the second lock, the second release message including data indicating the third value of the version number in the second lock and the***

exclusive lock type, wherein the third value of the version number is used by the lock manager to replace the second value of the version number in the lock data structure.

Regarding to claim 11, although Kavanagh further discloses the step of ***sending to the lock manager process a second release message to release the second lock*** (FIG. 38). However, claim 11 is a dependence of claim 9, which is objected to as being dependent upon objected claim 8. Thus, claim 11 is objected.

Conclusion

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Hung Pham whose telephone number is 703-605 4242. The examiner can normally be reached on Monday-Friday, 7:00 Am - 3:30 Pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, VU, KIM YEN can be reached on 703-305 4393. The fax phone numbers for the organization where this application or proceeding is assigned are 703-746 7239 for regular communications and 703-746 7238 for After Final communications. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703-305 3900.

Examiner: Hung Pham
February 10, 2003

KIM VU
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100